UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

CLAUDINEI MOREIRA DA SILVA

AVALIAÇÃO DE *DATASETS* E DE ALGORITMOS DE DETECÇÃO DE MUDANÇA UTILIZANDO MAPAS DE DIFICULDADE

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2021

CLAUDINEI MOREIRA DA SILVA

AVALIAÇÃO DE *DATASETS* E DE ALGORITMOS DE DETECÇÃO DE MUDANÇA UTILIZANDO MAPAS DE DIFICULDADE

Evaluation of Datasets and Change Detection Algorithms using Difficulty Maps

Dissertação apresentada ao Programa de Pós-Graduação em Informática – PPGI, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Silvio Ricardo Rodrigues Sanches

Co-orientador: Prof. Dr. Roberto Sadao Yokoyama

CORNÉLIO PROCÓPIO
2021

-

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio

CLAUDINEI MOREIRA DA SILVA

AVALIAÇÃO DE DATASETS E DE ALGORITMOS DE DETECÇÃO DE MUDANÇA UTILIZANDO MAPAS DE DIFICULDADE

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Informática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Computação Aplicada.

Data de aprovação: 26 de Fevereiro de 2021

Prof Roberto Sadao Yokoyama, Doutorado - Fundação Universidade Federal do Abc (Ufabc)

Prof Silvio Ricardo Rodrigues Sanches, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Antonio Carlos Sementille, Doutorado - Universidade Estadual Paulista Júlio de Mesquita Filho (Unesp)

Prof Claiton De Oliveira, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Cleber Gimenez Correa, Doutorado - Universidade Tecnológica Federal do Paraná

Dedico a realização deste trabalho:

Aos meus pais, Maria Nilza e Sebastião (ambos *In memorium*), que me deram a vida e a herança de bons valores e exemplos.

Aos meus irmãos, Sidnéia, Valdinei e Cleber pelo carinho e preocupação.

À Fabiana, pelo incentivo, apoio e compreensão desde o início.

Ao meu filho Felipe, por ser a grande força motora, cuja qual faz tudo ter sentido e faz valer a pena todo sacrifício.

AGRADECIMENTOS

A Deus, pela dádiva da vida e por me permitir realizar tantos projetos e sonhos nesta existência. Obrigado por me permitir errar, aprender e crescer, por Sua eterna compreensão e tolerância, por Seu infinito amor, pela coragem que não me permitiu desistir e principalmente por ter me dado uma família tão especial, enfim, obrigado por tudo.

Aos professores do Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná de Cornélio Procópio/PR, por terem tão pacientemente transferido o conhecimento base para que fosse possível o desenvolvimento deste trabalho.

À Fundação Educacional do Município de Assis – FEMA, representados pelos administradores e colegas de trabalho, por todo apoio.

Aos professores Silvio Sanches e Roberto Sadao, pela orientação, competência, profissionalismo e dedicação a este trabalho.

Ao CNPq pelo apoio financeiro – projeto CNPq/Universal (427485/2018-5)

“A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”. (Albert Einstein)

“Os que se encantam com a prática sem a ciência são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino”. (Leonardo da Vinci)

RESUMO

SILVA, Claudinei Moreira. AVALIAÇÃO DE *DATASETS* E DE ALGORITMOS DE DETECÇÃO DE MUDANÇA UTILIZANDO MAPAS DE DIFICULDADE. 55 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

A avaliação de um algoritmo de detecção de mudança deve mostrar a superioridade do seu desempenho em relação aos desempenhos dos algoritmos do estado-da-arte. As etapas da avaliação de um algoritmo consiste basicamente na sua execução para segmentar um conjunto de vídeos de um *dataset* e na comparação dos resultados com um *ground truth*. Neste trabalho, propõe-se a utilização de uma nova informação no processo de avaliação de algoritmos de detecção de mudança: o nível de dificuldade para classificar cada pixel de cada quadro dos vídeos de um *dataset*. Para cada quadro de vídeo, foi criado uma estruturada chamada mapa de dificuldade, que armazena valores que representam o nível de dificuldade exigido de um algoritmo para classificar cada pixel desse quadro. Baseado nesses mapas, foram desenvolvidas uma métrica que tem como objetivo avaliar o desempenho de algoritmos em relação ao mapa de dificuldade e outra que tem como objetivo estimar o nível de dificuldade que cada vídeo do *dataset* exige dos algoritmos para classificar os pixels de seus quadros. Um método para selecionar os vídeos representativos de um *dataset* também foi desenvolvido neste trabalho. Os resultados da aplicação da métrica para avaliação de algoritmos mostraram que os algoritmos que representam o estado-da-arte normalmente falham nas mesmas regiões do quadro. A aplicação da métrica que estima níveis de dificuldade de vídeos mostrou que muitos vídeos do *dataset* CDNet 2014 possuem níveis de dificuldade similares. Essa constatação corrobora com os resultados obtidos da aplicação do método para gerar um subconjunto representativo, uma vez que o subconjunto selecionado possui menos vídeos e apresenta o mesmo potencial de avaliação do conjunto de vídeos original.

Palavras-chave: Detecção de Mudança, *Dataset*, Métrica

ABSTRACT

SILVA, Claudinei Moreira. EVALUATION OF DATASETS AND CHANGE DETECTION ALGORITHMS USING DIFFICULTY MAPS. 55 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

Evaluating a change detection algorithm must show the superiority of its performance concerning state-of-the-art algorithms' performance. The steps of evaluating an algorithm comprise executing it to segment a set of videos from a *dataset* and comparing the results with ground truth. In this work, we propose using additional information in evaluating change detection algorithms: the level of difficulty to classify each pixel of each frame of the videos from a dataset. For each video frame, we created a structured called difficulty map, which stores values representing the level of difficulty required by an algorithm to classify each pixel of that frame. Based on the difficulty maps, we developed two metrics. The first aims to evaluate the performance of algorithms about the difficulty map. The second metric aims to estimate the level of difficulty that each dataset video requires from the algorithms to classify its frames' pixels. In this work, we also developed a method for selecting representative videos based on their difficulty level. The evaluation algorithms' results showed that the algorithms that represent the state-of-the-art fail in the same regions of the frame. The metric that estimates video difficulty levels has been demonstrated that many videos from dataset CDNet 2014 have similar difficulty levels. This finding corroborates the method's results to generate a representative subset since the selected subset has fewer videos and has the same evaluation potential as the original video set.

Keywords: Change Detection, Dataset, Metric

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Vigilância por vídeo, ambientes inteligentes e recuperação de conteúdo são exemplos de sistemas que utilizam algoritmos de detecção de mudança (*change detection*) na imagem monitorada (SANCHES et al., 2019). Tais algoritmos identificam regiões (conjunto de pixels) que sofrem modificações ou se movem em relação a uma imagem que representa o modelo do plano de fundo da cena (GOYETTE et al., 2012). A detecção de mudança é pré-requisito para muitas aplicações de visão computacional e processamento de vídeo (GOYETTE et al., 2012).

A avaliação do desempenho de um algoritmo de detecção de mudança é uma tarefa fundamental em que os autores devem mostrar claramente que o novo algoritmo apresentado é superior em desempenho aos encontrados na literatura. As etapas da avaliação de desempenho de um algoritmo consiste basicamente na sua execução para segmentar um conjunto de vídeos, chamado *dataset*, e comparar os resultados com um *ground truth*. Um *ground truth* é um conjunto de quadros rotulados manualmente por um especialista, que permite identificar o resultado ideal da segmentação. Dessa forma os resultados da segmentação dos algoritmos, que normalmente contêm erros de classificação de pixels, são obtidos e comparados com o *ground truth*. Em seguida, são calculadas métricas que representam o desempenho do algoritmo (*F-Measure*, Revocação, Precisão etc).

Quando vários algoritmos utilizam o mesmo *dataset* e as mesmas métricas no processo de avaliação, é possível comparar seus desempenhos, pois os resultados desses algoritmos são obtidos considerando as mesmas ferramentas e métodos. Essa forma tradicional de avaliar desempenho é bem aceita pela comunidade científica. No entanto, outras informações relevantes que podem ser úteis para comparar algoritmos também podem ser obtidas dos resultados da segmentação dos vídeos. O nível de

dificuldade para classificar determinado *pixel*, por exemplo, pode ser importante para identificar as regiões de um quadro nas quais é difícil diferenciar o que é elemento de interesse (região que ocorre mudança) do que é plano de fundo. Essas regiões são pixels isolados ou conjuntos de pixels pertencentes aos vídeos de *datasets* em que a maioria dos algoritmos falha ao classificá-los (SANCHES et al., 2019).

Um vez que os algoritmos de detecção de mudança se baseiam nas mais diferentes abordagens (redes neurais, algoritmos genéticos, lógica *fuzzy* etc) (SOBRAL; VACAVANT, 2014), uma solução pode ser capaz de classificar corretamente pixels difíceis de serem classificados – onde a maioria dos algoritmos falha – mas pode falhar em regiões que são classificados corretamente pelos algoritmos do estado-da-arte. Este algoritmo pode ser considerado promissor, pois a melhora de seu desempenho consiste em superar desafios que foram superados por alguns algoritmos da literatura.

Para avaliar um algoritmo de detecção de mudança é desejável que o conjunto de vídeos utilizado seja capaz de diferenciar um algoritmo eficiente de outro pouco eficiente. Para isso, os vídeos do conjunto utilizado devem conter vídeos que possuam níveis de dificuldade distintos. A obtenção do nível de dificuldade torna possível remover um vídeo que ofereça aos algoritmos dificuldade similar ao de outro do conjunto e, ao mesmo tempo, incluir um vídeo que ofereça aos algoritmos um nível de dificuldade diferente dos demais. Essa estratégia possibilita a construção de *datasets* com vídeos organizados em uma escala, sem redundância, de níveis de dificuldade. Consequentemente, cada vídeo fornece uma informação diferente sobre o desempenho de um algoritmo. Além disso, essa informação torna possível identificar subconjuntos representativos de vídeos, que têm potencial para avaliar algoritmos similar ao do *dataset* completo.

Considera-se que um subconjunto de vídeos tem potencial de avaliação similar ao conjunto completo quando: (i) avalia-se vários algoritmos utilizando esses dois conjuntos (todos os vídeos do *dataset* e subconjunto representativo) e (ii) a ordem dos algoritmos é a mesma nas duas avaliações quando esses algoritmos são ordenados considerando seus desempenhos.

Os subconjuntos representativos são úteis, por exemplo, para melhorar a

eficiência dos modelos de aprendizagem. Nesses casos, como não há redundância, a possibilidade de o modelo tomar decisões com base em ruídos é pequena. Além disso, um conjunto com poucos vídeos representativos pode ser importante durante o desenvolvimento de um novo algoritmo, uma vez que o desempenho de versões preliminares podem ser avaliadas com maior rapidez. Um subconjunto representativo não pode ser gerado quando os vídeos do conjunto original possuem níveis de dificuldade distintos. Nesse caso, o método para gerar subconjuntos representativos pode ser utilizado para avaliar se um novo vídeo que pretende-se incluir no *dataset* possui nível de dificuldade distinto dos demais, de forma que contribua com o conjunto original na avaliação de algoritmos.

1.1 OBJETIVOS

Os objetivos deste trabalho são (i) desenvolver uma métrica para identificar algoritmos de detecção de mudança promissores – algoritmos que classificam corretamente os pixels que a maioria dos algoritmos do estado-da-arte não são capazes de classificar, (ii) desenvolver uma métrica para identificar o nível de dificuldade que os vídeos de um *dataset* exige dos algoritmos de detecção de mudança para classificar corretamente os pixels de seus quadros e (iii) desenvolver um método para selecionar um subconjunto representativo, sem redundância de vídeos e com o mesmo potencial para avaliar algoritmos que o conjunto de vídeos original.

1.2 ORGANIZAÇÃO DO DOCUMENTO

Os demais capítulos deste trabalho estão organizados da forma como segue. O Capítulo 2 apresenta os *datasets* mais utilizados pelos pesquisadores para avaliar algoritmos de detecção de mudança. As métricas propostas para avaliar algoritmos e estimar os níveis de dificuldade de vídeos de *datasets* são descritas no Capítulo 3. No Capítulo 4 são apresentados os experimentos realizados e os resultados obtidos da aplicação das métricas. Finalmente, reservou-se o Capítulo 5 para apresentar as conclusões e as perspectivas de trabalhos futuros.

2 FUNDAMENTOS DE AVALIAÇÃO DE *DATASETS* E TRABALHOS RELACIONADOS

Para demonstrar o desempenho de seus algoritmos, alguns autores adotam a estratégia de criar seus próprios vídeos e *ground truths*. Essa estratégia, no entanto, não permite comparar o desempenho do novo algoritmo com o desempenho dos algoritmos que representam o estado-da-arte. Para facilitar essa comparação, a maioria dos autores utiliza *datasets* (SANCHES et al., 2021). Nesse capítulo são apresentados as fases e os recursos necessários para avaliar algoritmos, incluindo os principais *datasets* disponíveis na literatura e métricas utilizadas para representar o desempenho de algoritmos.

A avaliação de um novo algoritmo de detecção de mudança compreende 4 etapas: (i) executar o algoritmo para segmentar vídeos de um *dataset*, (ii) comparar os resultados da segmentação com um *ground truth*, (iii) calcular uma métrica ou um conjunto de métricas que representam o desempenho do algoritmo, e (iv) comparar o desempenho do novo algoritmo com os desempenhos dos algoritmos que representam o estado-da-arte. Esse tipo de avaliação é aplicada tanto durante a fase de desenvolvimento, para testar versões intermediárias, quanto depois de finalizado o desenvolvimento.

O recurso mais importante no processo de avaliação de algoritmos é o *dataset*. Os *datasets* normalmente possuem conjuntos de vídeos, *ground truths*, resultados de algoritmos que representam o estado-da-arte e ferramentas de auxílio aos pesquisadores. A utilização de *datasets* permite a comparação do desempenho de vários algoritmos uma vez que algoritmos diferentes são executados e produzem resultados da segmentação considerando um mesmo conjunto de vídeos (SANCHES et al., 2019). Um *dataset* para avaliar algoritmos de detecção de mudanças normalmente contém vídeos com cenas que simulam situações típicas do ambiente

de uma determinada aplicação (UNIVERSITÉ DE SHERBROOKE, 2019).

A Figura 1 mostra exemplos de quadros de vídeos de *datasets* utilizados para avaliar algoritmos de detecção de mudança. Na Figura 1a é exibido um quadro que pertence a um vídeo cuja cena é comum em aplicações como contagem de pessoas e detecção de quedas de pessoas. A Figura 1b mostra uma cena comum de aplicações que detectam objetos abandonados ou identificam tumultos em área movimentadas. Os quadros exibidos nas Figuras 1c e 1d representam cenas típicas de aplicações de reconhecimento facial e contagem de veículos, por exemplo.

(a) (b)

(c) (d)

Figura 1: Quadros de vídeos exibindo cenas típicas de aplicações que utilizam algoritmos de detecção de mudança. (a) contagem de pessoas e detecção de quedas de pessoas (WANG et al., 2014b), (b) detecção de objetos abandonados e identificação de tumultos em área movimentadas (WANG et al., 2014b), (c) reconhecimento facial (TOYAMA et al., 1999) e (d) detecção de acidentes e contagem de veículos (WANG et al., 2014b)

A geração de um *dataset* é um processo trabalhoso, principalmente em função do esforço necessário para criação do *ground truth*. Para cada quadro do vídeo original deve ser gerado um quadro correspondente em que os pixels que pertencem

ao elemento de interesse são rotulados com uma determinada cor, geralmente a cor branca, e os pixels que pertencem ao plano de fundo são rotulados com uma cor diferente, normalmente a cor preta. A atribuição de rótulos é feita quase sempre de forma manual.

Existem ainda *datasets* em que são rotuladas outras regiões do quadro, por exemplo, sombras, regiões fora da área de interesse na cena e regiões indeterminadas, onde não é possível identificar visualmente se o pixel pertence ao fundo ou ao elemento de interesse (ocorre com frequência nas bordas do elemento de interesse). Os pixels dessas regiões são rotulados com cores diferentes das atribuídas ao fundo e ao elemento de interesse (geralmente tons de cinza). A Fig. 2 mostra o vídeo "highway" obtido do *dataset* CDNet2014 (WANG et al., 2014b) e seus respectivos *ground truths*.

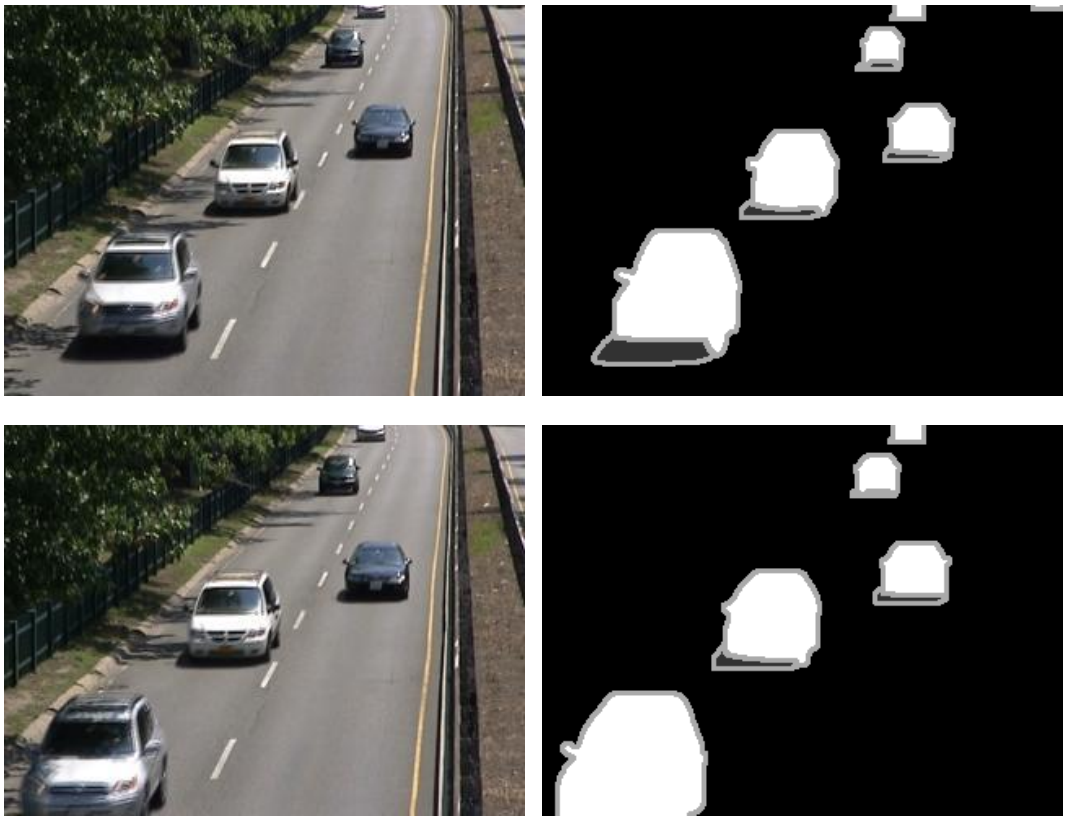

Figura 2: Vídeo "highway" obtido do *dataset* CDNet2014 (WANG et al., 2014b) e seus respectivos *ground truths*

2.1 PRINCIPAIS *DATASETS* DISPONÍVEIS PARA AVALIAÇÃO DE DESEMPENHO

Existem *datasets*, como o *Performance Evaluation of Tracking and Surveillance* (PETS) (Young; Ferryman, 2005) e o *CAVIAR Test Case Scenarios* (FISHER, 2019), que são utilizados para a avaliação de desempenho de algoritmos de rastreamento. Esses *datasets* não são adequados para avaliar algoritmos de detecção de mudança uma vez que os rótulos dos *ground truths* são *bounding boxes*. Nesse caso, as regiões do fundo dentro de um *bounding box* são tratadas como pertencentes ao elemento de interesse, o que não representa um problema quando se avalia algoritmos de rastreamento. No entanto, vídeos de *datasets* para avaliar algoritmos de detecção de mudança devem ser rotulados de forma mais precisa, considerando o contorno do elemento de interesse, como mostrado na Figura 2.

Datasets como o *Scene Background Modeling* (SBMnet) são utilizados para avaliação de desempenho de algoritmos que geram um modelo do fundo que tem a finalidade de inicializar algoritmos de subtração de fundo (UNIVERSITY OF NAPLES PARTHENOPE, 2019). O problema da geração desse modelo é chamado de *bootstrapping*. Nesses *datasets*, os *ground truths* são imagens de fundo sem qualquer elemento de interesse presente na cena. Nos *datasets* utilizados para avaliação de desempenho de algoritmos de detecção de mudança, cada quadro do vídeo original tem um quadro correspondente no qual os pixels pertencentes ao elemento de interesse são rotulados com uma cor particular, geralmente branco, e os pertencentes ao fundo são rotulados com um cor diferente, geralmente preta. Os pixels das regiões como sombra, região desconhecida, por exemplo, são rotulados com níveis de cinza.

O *Background Models Challenge* (BMC) (VACAVANT et al., 2013) é um *dataset* bastante utilizado por pesquisadores da área. O BMC contém 29 vídeos para avaliação de desempenho de algoritmos de detecção de mudança (20 vídeos sintéticos e 9 reais). A utilização de vídeos sintéticos (gerados por computação gráfica) facilitam a geração do *ground truth*, pois os pixels pertencentes ao elemento de interesse e ao fundo são conhecidos do sistema. No entanto, esses vídeos não reproduzem a naturalidade das cenas reais, o que pode comprometer a avaliação do

desempenho dos algoritmos (SANCHES et al., 2019).

Todos os vídeos do BMC possuem *ground truth* (Kalsotra; Arora, 2019). O conteúdo dos vídeos apresentam cenas com desafios para os algoritmos, como variações na iluminação, elemento de interesse ocupando grande parte da cena e plano de fundo dinâmico (Kalsotra; Arora, 2019). O BMC disponibiliza uma ferramenta gráfica, chamada “BMC Wizard”, que auxilia o pesquisador na utilização do *dataset*. Por meio da ferramenta é possível usar os vídeos e seus *ground truths* para visualizar os resultados e calcular métricas (Revocação, Precisão, *F-Measure*, etc) que representam o desempenho do algoritmo em avaliação. Apesar de muito utilizado, os vídeos do BMC deixaram de ser disponibilizados *online* pelos autores, assim como ocorreu com o *dataset* I2R (LI et al., 2004), que possuía 9 vídeos – em que apenas alguns de seus quadros continham *ground truths* – e foi bastante utilizado há alguns anos.

O Wallflower (TOYAMA et al., 1999; MICROSOFT CORPORATION, 2019) também é um conjunto de vídeos bastante utilizado para avaliar algoritmos de detecção de mudança, ainda que seus vídeos sejam de baixa resolução e que apenas um quadro de cada um deles tenha rótulos. Essas limitações e a pouca quantidade de vídeos oferecidos diminuem o potencial de avaliação do WallFlower. Esse *dataset* contém 7 vídeos que apresentam cenas com alguns desafios, como variações de iluminação, plano de fundo dinâmico e camuflagem (Kalsotra; Arora, 2019). Ao contrário do BMC, o WallFlower não disponibiliza ferramentas que auxiliem a avaliação de desempenho de algoritmos.

O *dataset* mais utilizado pela comunidade científica para avaliação de desempenho de algoritmos de detecção de mudança é o ChangeDetection.net (CDNet) (UNIVERSITÉ DE SHERBROOKE, 2019). O CDNet possui duas versões: o CDNet 2012 (GOYETTE et al., 2012) que contém 31 vídeos e o CDNet 2014 que contém 53 vídeos. Todos os quadros do *ground truth* do CDNet são rotulados e, além disso, uma característica que diferencia o CDNet é que os quadros dos seus vídeos são rotulados com regiões de sombra, além das regiões do fundo e do elemento de interesse. Existem ainda vídeos capturados por câmeras PTZ, câmeras IP de baixa resolução e câmeras térmicas (UNIVERSITÉ DE SHERBROOKE, 2019).

Os 31 vídeos do CDNet 2012 são divididos em 6 categorias para cobrir uma gama de desafios que existem na maioria das aplicações que envolvem análise de vídeo (Kalsotra; Arora, 2019).

No CDNet 2014, além dos vídeos da versão 2012 foram incluídas 5 novas categorias de vídeos que apresentam cenas complexas, cada uma simulando um tipo de desafio diferente (Kalsotra; Arora, 2019) para os algoritmos. Tanto o CDNet 2012 quanto o CDNet 2014 disponibilizam ferramentas nas linguagens Matlab e Python para auxiliar o pesquisador a avaliar o desempenho de um novo algoritmo. Nesta pesquisa, foram utilizados os vídeos do CDNet 2014 para validar nossas métricas e método.

Na Tabela 1 os *datasets* analisados neste levantamento são comparados de acordo com as seguintes características: a existência de ferramenta de auxílio para a utilização do *dataset*, o número de vídeos que compõem o *dataset*, o vídeo com resolução mais baixa, o vídeo com a resolução mais alta, o menor número de quadros em um vídeo, o maior número de quadros em um vídeo e a quantidade de quadros em cada vídeo que possuem *ground truth*.

Tabela 1: Características dos principais *datasets* utilizados para avaliar algoritmos de detecção de mudanças

Dataset	Ferr.	Núm.	Resolução		Núm. quadros		Quadros
		vídeos	Mín	Máx	Mín	Máx	rotulados
Wallflower (TOYAMA et al., 1999)	Não	7	160x120	160x120	286	5889	1
BMC (VACAVANT et al., 2013)	Sim	9*	320x240	320x240	793	117151	todos
I2R (LI et al., 2004)	Não	9	176x144	176x144	633	23893	20
CDNet2012 (GOYETTE et al., 2012)	Sim	31	320x240	720x546	600	7999	todos
CDNet2014 (WANG et al., 2014b)	Sim	53	320x240	720x546	600	7999	todos

*número de vídeos reais (BMC possui alguns vídeos sintéticos)

Como pode ser observado, o CDNet 2014 mostra-se o mais adequado para avaliar o desempenho de algoritmos, uma vez que disponibilizam mais vídeos, de mais alta resolução e com todos os quadros rotulados.

2.2 MÉTRICAS QUE REPRESENTAM O DESEMPENHO DOS ALGORITMOS

Outro componente importante na avaliação de algoritmos de detecção de mudança são as métricas. Muitos *datasets* oferecem aos pesquisadores ferramentas que calculam essas métricas e os resultados obtidos são utilizados para comparar o

novo algoritmo com o estado-da-arte.

Considerando valores normalizados, os algoritmos de detecção de mudança geram uma mascára $S \in \{0,1\}^{n \times m}$ como resultado, onde 0 é o rótulo dos pixels classificados como fundo, 1 é o rótulo dos pixels classificados como elemento de interesse e $n \times m$ é o tamanho do quadro. Os rótulos do *ground truth* $G \in [0,1]^{n \times m}$ são 0 e 1, que representam os pixels pertencentes ao plano do fundo e os pertencentes a um elemento de interesse, respectivamente. Além disso, existem rótulos para regiões como sombras, regiões fora da área de interesse na cena e regiões desconhecidas (UNIVERSITÉ DE SHERBROOKE, 2019).

Dada a norma da matriz $\|A\| = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|$, os verdadeiros positivos (pixels classificados corretamente como pertencentes ao elemento de interesse) são definidos como $TP = \|G \odot S\|$, onde $\odot$ é o produto *entrywise*. Os falsos positivos (pixels do plano de fundo classificados incorretamente como pertencentes a um elemento de interesse) são definos como $FP = \|(1-G) \odot S\|$, os verdadeiros negativos (pixels do plano de fundo classificados corretamente como pertencentes ao plano de fundo) são definidos como $TN = \|(1-G) \odot (1-S)\|$ e os falsos negativos (pixels do elemento de interesse incorretamente classificados como pertencentes ao plano de fundo) são definidos como $FN = \|G \odot (1-S)\|$.

As métricas taxa de falsos positivos (*False Positive Rate* – FPR) e taxa de falsos negativos (*False Negative Rate* – FNR), por exemplo, podem ser calculadas de acordo com as equações 1 e 2.

$$FPR = \frac{FP}{FP + TN} \tag{1}$$

$$FNR = \frac{FN}{TP + FN}. \tag{2}$$

A métrica Precisão (Pr) é calculada de acordo com a equação

$$Pr = \frac{TP}{TP + FP} \tag{3}$$

e a métrica Revocação (Re) é definida pela equação

$$Re = \frac{TP}{TP+FN}. \tag{4}$$

Outras métricas utilizadas para avaliar o desempenho de algoritmos de detecção de mudanças é a Acurácia (Ac), definida como

$$Ac = \frac{TP+TN}{TP+TN+FP+FN}, \tag{5}$$

a Especificidade (Sp), definida como

$$Sp = \frac{TN}{TN+FP}, \tag{6}$$

e o Percentual de Erros de Classificação (*Percentage of Wrong Classifications* – PWC)

$$PWC = \frac{100 \times (FN+FP)}{TP+FN+FP+TN}. \tag{7}$$

A *F-Measure* ($F1$) é a métrica mais utilizada para representar o desempenho de algoritmos de detecção de mudança (GOYETTE et al., 2012; UNIVERSITÉ DE SHERBROOKE, 2019). A $F1$, que é baseada nas métricas Pr e Re é calculada pela equação

$$F1 = \frac{2 \times Pr \times Re}{Pr+Re}. \tag{8}$$

3 UMA ABORDAGEM COM MAPAS DE DIFICULDADE PARA AVALIAÇÃO DE *DATASET* E DE ALGORITMOS DE DETECÇÃO DE MUDANÇA

A abordagem proposta consiste em inicialmente gerar uma estrutura chamada mapa de dificuldade, que armazena o nível de dificuldade exigido para um algoritmo classificar corretamente os pixels de um quadro de vídeo. Esse mapa pode ser utilizado como um recurso para avaliar novos algoritmos de detecção de mudança ou para estimar o nível de dificuldade que um vídeo exige dos algoritmos para classificar corretamente seus pixels. Primeiramente, descreve-se neste capítulo o processo de geração de um mapa de dificuldade e, em seguida, apresenta-se as formas de utilizar um mapa para suas duas finalidades: avaliação de algoritmos e estimação de nível de dificuldade de vídeos. O método para selecionar um subconjunto representativo do *dataset* é apresentado no final deste Capítulo.

3.1 GERAÇÃO DO MAPA DE DIFICULDADE

Uma estrutura chamada mapa de dificuldade, que é a informação base das métricas propostas neste trabalho, deve ser gerada utilizando os resultados de diversos algoritmos, preferencialmente os que representam o estado-da-arte. Com esses resultados, é possível identificar o nível de dificuldade de um pixel contando quantos algoritmos do grupo classificaram incorretamente esse pixel. Dessa forma, para cada quadro de um vídeo, gera-se um mapa de dificuldade correspondente, que é responsável por armazenar esses valores.

Considerando valores normalizados, um algoritmo de detecção de mudança gera uma máscara $S \in \{0,1\}^{l \times c}$ como resultado, onde 1 é o rótulo dos pixels da região de interesse (áreas em que houve mudança), 0 é rótulo dos pixels do plano de fundo e $l \times c$ é a resolução do quadro do video (linha $\times$ coluna). Os rótulos do *ground truth*

$G \in [0,1]^{l \times c}$ são o valor 0 (pixels pertencentes ao plano de fundo) e o valor 1 (pixels pertencentes a um elemento de interesse).

O *ground truth* também pode possuir rótulos para regiões como “sombras” e “regiões indeterminadas”, que estão localizadas normalmente ao redor de elementos de interesse (UNIVERSITÉ DE SHERBROOKE, 2019). Essa é a região onde não é possível visualmente identificar se os pixels pertencem ao fundo ou ao elemento de interesse. A matriz $R \in \{0,1\}^{l \times c}$ foi definida para indicar os pixels que pertencem à região de interesse do *ground truth*, que são apenas os pixels rotulados como elemento de interesse ou plano de fundo.

Para gerar um mapa de dificuldade, a abordagem proposta consiste em executar vários algoritmos para segmentar vídeos de um *dataset*. Em seguida, as máscaras S que contêm os resultados dos algoritmos são comparadas com o *ground truth* G para identificar os pixels classificados incorretamente. O nível de dificuldade de um pixel é dado pelo número de algoritmos que classificaram incorretamente o pixel. Para cada quadro de cada vídeo de um *dataset* é gerado um mapa de dificuldade, que é definido como $D \in [0,1]^{l \times c}$ e armazena o nível de dificuldade para classificar cada pixel.

O número de algoritmos utilizado determina a quantidade de níveis de dificuldade contidos no mapa. Para n algoritmos, $n+1$ níveis de dificuldade são representados no mapa utilizando diferentes tons da cinza. A Figura 3 apresenta um exemplo de um quadro que pertence a um *ground truth* G, um quadro de uma matriz R (que indica os pixels de interesse) e 4 quadros com os resultados de 4 diferentes algoritmos ($n = 4$) de detecção de mudança (máscaras S).

As regiões delimitadas pelos retângulos vermelhos dentro do *ground truth*, da matriz R, das máscaras S e do mapa de dificuldade são apresentadas numericamente pelas matrizes posicionadas acima de cada um desses elementos. Nas matrizes correspondentes às máscaras (algoritmos 1, 2, 3 e 4), os pixels vermelhos representam os erros dos algoritmos, os pixels verdes representam os acertos dos algoritmos e o pixel cinza representa uma região indeterminada, que será desconsiderada na geração do mapa.

Nas máscaras S, os algoritmos atribuíram a cor branca aos pixeis que classificaram como pertencentes aos elementos de interesse e a cor preta aos pixels que classificaram como pertencentes ao plano de fundo da cena contida no quadro. No *ground truth*, onde os rótulos são atribuídos manualmente, existe ainda o rótulo cinza, que representa a região indeterminada. Apenas os pixels rotulados com o valor 1 na matriz R (região de interesse) são considerados.

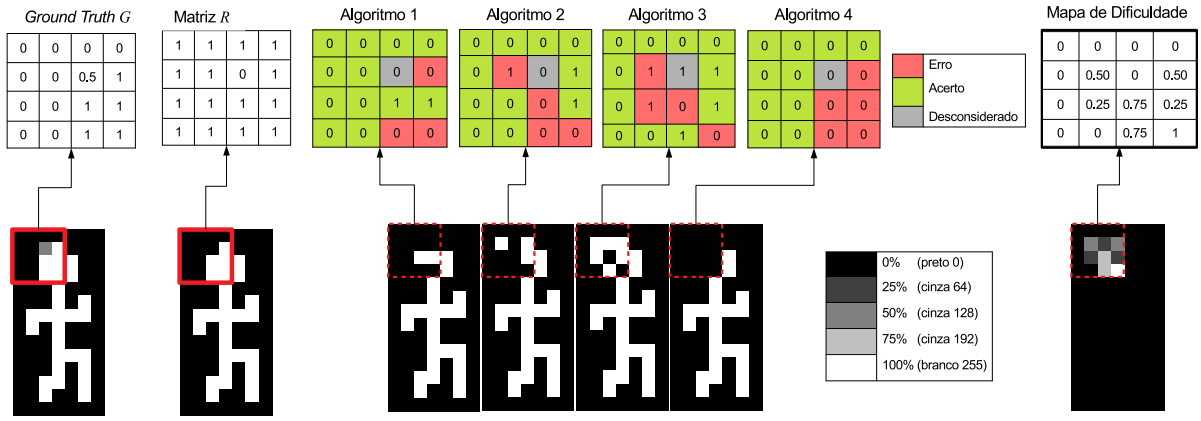

Figura 3: Um mapa de dificuldade com $n = 4$. Neste exemplo, o *ground truth* tem pixels rotulados como plano de fundo (rótulo = 0), primeiro plano (rótulo = 1) e região indeterminada (rótulo = 0.5)

No exemplo, o mapa de dificuldade que utiliza os resultados dos 4 algoritmos é constituído por 5 níveis de dificuldade $(n + 1)$, sendo, nível 0 (pixel com nenhuma dificuldade) e 1 (pixel com a mais alta dificuldade). Além disso, a dificuldade atribuída aos demais pixels é determinada pela porcentagem de algoritmos que erraram sua classificação. No exemplo da Figura 3, quando apenas um algoritmo falha (25%) o pixel é rotulado com o nível 0.25, quando 3 algoritmos falham (75%) rotula-se o pixel com o nível 0.75 e assim, sucessivamente. Cada nível no exemplo é representado por uma cor diferente, 0% (preto 0), 25% (cinza 64), 50% (cinza 128), 75% (cinza 192) e 100% (branco 255). O Algoritmo 1 mostra o pseudocódigo que implementa a abordagem mostrada na Figura 3.

Algoritmo 1 Pseudocódigo para gerar Mapas de Dificuldade

Entradas: S (resultado da segmentação), R (região de interesse), G (*ground truth*), V (número de vídeos), Q (número de quadros), $(l \times c)$ (número de pixels) e n (número de algoritmos)
Saídas: D (estrutura que armazena mapas de dificuldade)

$D_{vid.frame.pixel.level=0}$
for i= 1 **to** V **do**
  for j= 1 **to** Q **do**
    for k=1 **to** $(l \times c)$ **do**
      for m=1 **to** n **do**
        if $S_{vid(i).frame(j).pixel(k).alg(m).label} != G_{vid(i).frame(j).pixel(k).alg(m).label}$ **and**
          $R_{vid(i).frame(j).pixel(k).label} == 1$ **then**
          $D_{vid(i).frame(j).pixel(k).level}$ = $D_{vid(i).frame(j).pixel(k).level}$+1

O Matlab (R2013b) foi a ferramenta utilizada para implementação dos algoritmos desenvolvidos nesta pesquisa, inclusive o responsável pela geração dos mapas.

3.2 MÉTRICA PARA AVALIAÇÃO DE ALGORITMOS

Um mapa de dificuldade pode ser utilizado como uma ferramenta que auxilia a obtenção de uma nova medida de desempenho de um algoritmo. Avaliar um algoritmo por meio de um mapa consiste em comparar os quadros desse mapa com o *ground truth* G e com os resultados dos algoritmos, apresentados na forma de máscaras S. O *ground truth* é necessário para que seja identificada a região do quadro em que o erro ocorreu (elemento de interesse ou plano de fundo), pois essa informação não está contida no mapa. O mapa de dificuldade armazena a frequência de erros dos algoritmos em um determinado pixel, sem especificar a região que o pixel pertence.

O *ground truth* permite que os falsos positivos FP sejam diferenciados dos falsos negativos FN e que os verdadeiros positivos TP sejam diferenciados dos verdadeiros negativos TN. Uma vez identificado o tipo do erro, os níveis de dificuldade dos pixels que estão armazenados no mapa possibilitam calcular os valores TP_D, TN_D, FP_D e FN_D por meio das equações

$$TP_D = \|G \odot S \odot D \odot R\|, \tag{9}$$

$$TN_D = \|(1-G) \odot (1-S) \odot D \odot R\|, \tag{10}$$

$$FP_D = \|(1-G) \odot S \odot D \odot R\| \tag{11}$$

e

$$FN_D = \|G \odot (1-S) \odot D \odot R\| \tag{12}$$

TP_D, TN_D, FP_D e FN_D são utilizados para calcular a Taxa de Falsos Positivos (FPR_D), a Taxa de Falsos Negativos (FNR_D), Precisão (Pr_D), Revocação (Re_D), Especificidade (Sp_D), Percentual de Classificações Incorretas (PWC_D) e F-Measure ($F1_D$). Essas métricas representam o desempenho de um algoritmo de detecção de mudança em relação ao mapa de dificuldade.

3.3 MÉTRICA PARA ESTIMAR O NÍVEL DE DIFICULDADE DE UM VÍDEO

Além da avaliação de algoritmos, um mapa de dificuldade também pode ser utilizado para avaliar vídeos de um *dataset*, mais especificamente, para identificar os níveis de dificuldade desses vídeos. Esse nível de dificuldade está relacionado com o esforço necessário para classificar os pixels dos quadros desse vídeo. O conteúdo da cena define o esforço que um vídeo exige de um algoritmo.

Utilizando as informações dos mapas de dificuldades, é possível estimar o nível de dificuldade L de um vídeo para avaliar algoritmos. Esse valor pode ser utilizado para organizar vídeos de *datasets* de acordo com essa característica. *Datasets* que contenham vídeos com diferentes valores de L podem diferenciar com maior precisão os algoritmos mais eficientes dos menos eficientes.

Dado o número de pixels válidos Nvp para o j^{th} quadro de um *ground truth* G

$$Nvp_j = \sum_{i=1}^{l*c} p(i) \tag{13}$$

$$p(i) = \begin{cases} 1, & \text{if } R_{(i)} == 1 \\ 0, & \text{if } R_{(i)} == 0 \end{cases} \tag{14}$$

o nível de dificuldade L de uma sequência de quadros k pode ser obtido de acordo com a equação

$$L(k) = \sum_{j=start}^{end} \sum_{i=1}^{N_{vp}} f \times \frac{d(i,j,D_k)}{N_{vp_j}} \tag{15}$$

onde $start$ é o quadro inicial, end é o quadro final, D_k é o mapa de dificuldade da sequência de quadros k, f é uma constante para reduzir o tamanho da escala dos valores que representam o nível de dificuldade (definido empiricamente como 0,1) e $d(i,j,D_k)$ é o nível de dificuldade armazenado do pixel i do quadro j do mapa de dificuldade D_k.

O valor L pode ser calculado para um conjunto de quadros, para um vídeo completo ou para um *dataset*. Alguns *datasets* agrupam seus vídeos de acordo com algum desafio específico (por exemplo, plano de fundo dinâmico, sombras e tremulação da câmera) (UNIVERSITÉ DE SHERBROOKE, 2019). O nível de dificuldade também pode ser calculado para cada um desses grupos.

3.4 MÉTODO PARA SELECIONAR VÍDEOS REPRESENTATIVOS DE UM *DATASET*

Uma vez estimados os níveis de dificuldade L de cada vídeo do *dataset* é possível utilizar essa medida para avaliar se os vídeos originais formam um conjunto equilibrado no que se refere aos seus valores de L. Vídeos com níveis similares de dificuldade podem produzir o mesmo valor de desempenho quando um mesmo algoritmo segmenta seus quadros, ao passo que vídeos com valores de L distintos podem produzir diferentes valores para o desempenho desse mesmo algoritmo. Um *dataset* em que os vídeos são selecionados de forma que seus valores de L representem uma escala de dificuldade pode tornar mais precisa a avaliação e a comparação de algoritmos com desempenhos similares.

A métrica para avaliar vídeos de *datasets* desenvolvida nesta pesquisa possibilita o desenvolvimento de um método capaz de selecionar um subconjunto

representativo, que possui menos vídeos e tem o potencial de avaliação similar ao do conjunto original. Para isso, inicialmente, os vídeos do *dataset* devem ser ordenados de acordo com seus níveis de dificuldade. Em seguida, um vetor de distâncias M, que armazena as distâncias entre dois níveis consecutivos, deve ser preenchido.

Para encontrar e remover *outliers* desse vetor, o método de intervalo interquartil (IRQ) (RUSSEL; COHN, 2013) foi utilizado, gerando o vetor M'. A metade do valor da mediana calculada a partir dos valores do vetor M' representa o limiar t, que é utilizado para selecionar os vídeos que fazem parte do subconjunto representativo Rp. Considerando o conjunto de vídeos ordenados em ordem decrescente, um vídeo é incluído em Rp apenas quando a diferença entre o nível de dificuldade do próprio vídeo e o nível de dificuldade do vídeo seguinte for maior que t. Todo o processo de geração do subconjunto Rp pode ser visualizado no Algoritmo 2.

Algoritmo 2 Geração do subconjunto representativo, que possui potencial de avaliação similar ao do conjunto de vídeos original do *dataset*

Entradas: V (número de vídeos do *dataset*), v (conjunto de vídeos do *dataset*), L (níveis de dificuldade dos vídeos em ordem decrescente)
Saída: Rp (subconjunto representativo dos vídeos do *dataset*)

1: $Rp = 0$
2: **for** i=2 **to** V **do**
3: $M_{(i)}$ = $abs(L_{(i)}$ - $L_{(i-1)})$
4: $M' = M - outliers$
5: $t = (median(M'))/2$
6: **for** i=1 **to** $V - 1$ **do**
7: **if** $M_{(i)} > t$ **then**
8: $Rp = Rp + v(i)$

4 EXPERIMENTOS E RESULTADOS DAS AVALIAÇÕES DOS ALGORITMOS E DOS VÍDEOS USANDO MAPA DE DIFICULDADES

Vídeos contendo cenas de uma aplicação que utiliza algoritmos de detecção de mudança, os *ground truths* desses vídeos e os resultados da segmentação desses vídeos produzidos por algoritmos do estado-da-arte (máscaras S) são necessários para gerar um mapa de dificuldade utilizado a abordagem proposta (Algoritmo 1). Para avaliar o desempenho das novas métricas e do método para selecionar um subconjunto de vídeos representativos desenvolvidos neste trabalho são necessários também os resultados de algoritmos de detecção de mudanças diferentes dos utilizados para gerar os mapas de dificuldade. Todos esses recursos foram obtidos do *site* do CDNet 2014, conforme detalhado nas seções seguintes.

4.1 GERAÇÃO DOS MAPAS DE DIFICULDADE DOS VÍDEOS DO CDNET 2014

Pode-se obter um conjunto de máscaras S (resultados da segmentação) quando vários algoritmos segmentam vídeos de um *dataset*. Existem vários algoritmos de detecção de mudança que possuem o código-fonte disponibilizado pelos autores (ISIK et al., 2018). Existem também algoritmos tradicionais implementados em bibliotecas de software (OPENCV TEAM, 2019; SOBRAL; VACAVANT, 2014) ou ferramentas como o Matlab. Os resultados da segmentação de vários algoritmos de detecção de mudança podem ser obtidos utilizando essas implementações.

Neste trabalho, além dos vídeos e *ground truths*, foram utilizadas as máscaras S disponíveis no *site* do *dataset* CDNet 2014 (UNIVERSITÉ DE SHERBROOKE, 2019). No *site*, é disponibilizado também um *ranking* que mostra o desempenho de vários algoritmos de detecção de mudança. A Tabela 2 mostra os vídeos do CDNet 2014 e suas categorias. Cada uma das categorias representa um desafio

presente no conteúdo na cena (plano de fundo dinâmico, sombras etc) ou ocorrências que dificultam a ação do algoritmo de detecção de mudança (tremulação da câmera, vídeos com baixa taxa de quadros etc).

Tabela 2: Categorias e vídeos do CDnet 2014

Categorias	Vídeos
Baseline	highway, office, pedestrians e PETS2006
Dynamic Background	boats, canoe, fountain01, fountain02, overpass e fall
Camera Jitter	badminton, boulevard, sidewalk e traffic
Shadows	backdoor, bungalows, busStation, cubicle, peopleInShade e copyMachine
Interm. Object Motion	abandonedBox, parking, streetLight, sofa, tramstop e winterDriveway
Thermal	corridor, library, park, diningRoom e lakeSide
Bad Weather	blizzard, skating, snowFall e wetSnow
Low Framerate	port_0_17fps, tramCrossroad_1fps, tunnelExit_0_35fps e turnpike_0_5fps
Night Videos	bridgeEntry, busyBoulvard, fluidHighway, streetCorner, tramStation e winterStreet
PTZ	continuousPan, intermittentPan, twoPositionPTZCam e zoomInZoomOut
Turbulence	turbulence0, turbulence1, turbulence2 e turbulence3

O *ranking* do CDNet 2014 apresenta os algoritmos que obtiveram os melhores desempenhos entre os que utilizaram seus vídeos para avaliação. As métricas PFR, FNR, Pr, Re, Sp, PWC e $F1$ são utilizadas para representar os desempenhos nesse *ranking*. Segundo o levantamento realizado em Sanches et al. (2021), a $F1$ tem sido a métrica mais utilizada pelos pesquisadores.

Entre os algoritmos listados no *ranking* na data de acesso ao *site* (5 de agosto de 2019), algumas máscaras não estavam disponíveis para *download* (*links* quebrados). Um total de 42 algoritmos foram utilizados nesta pesquisa e as máscaras S de 30 deles, escolhidos aleatoriamente e listados na Tabela 3, foram utilizadas para geração dos mapas.

Conforme discutido e exemplificado na seção 3.1, um mapa gerado utilizando 30 algoritmos ($n = 30$) contém 31 níveis de dificuldade ($n + 1$). Para facilitar a visualização na forma de imagens, optou-se por armazenar cada nível de dificuldade nos mapas em intervalos de 8 valores, sendo que cada nível equivale a um tom

Tabela 3: Algoritmos utilizados para gerar os mapas de dificuldade

Algoritmo	Referência
WeSamBE	Jiang e Lu (2018)
SuBSENSE	St-Charles et al. (2015b)
SharedModel	Chen et al. (2015)
FTSG	Wang et al. (2014a)
CwisarDRP	Gregorio e Giordano (2017)
C-EFIC	Allebosch et al. (2016)
Multimode BS	Sajid e Cheung (2017)
EFIC	Allebosch et al. (2015)
CwisarDH	Gregorio e Giordano (2014)
Multimode BS Version 0	Maddalena e Petrosino (2010)
Spectral-360	Sedky et al. (2014)
SBBS	Varghese e G (2017)
BMOG	Martins et al. (2017)
AAPSA	Ramírez-Alonso e Chacon-Murguia (2016)
IUTIS-1	Bianco et al. (2017b)
GraphCutDiff	Miron e Badii (2015)
Mahalanobis distance	Benezeth et al. (2010)
SC_SOBS	Maddalena e Petrosino (2012)
RMoG	Varadarajan et al. (2013)
KDE - ElGammal	Elgammal et al. (2000)
GMM - Stauffer & Grimson	Stauffer e Grimson (1999)
CP3-online	Liang et al. (2015)
GMM - Zivkovic	Zivkovic (2004)
Euclidean distance	Benezeth et al. (2010)
BSPVGAN	Zheng et al. (2019)
FgSegNet-S (FPM)	Lim e Keles (2018)
IUTIS-3	Bianco et al. (2017b)
IUTIS-5	Bianco et al. (2017a)
PAWCS	St-Charles et al. (2015a)
WisenetMD	Lee et al. (2019)

de cinza (nível 0 = tom de cinza 0, nível 1 = tom de cinza 8. nível 2 = tom de cinza 16 ... nível 30 = tom de cinza 240). Uma imagem cujos valores dos pixels variam entre 0 (pixel sem dificuldade) e 240 (pixel com dificuldade máxima) representa um quadro de um mapa. Para facilitar os cálculos, os valores que representam os níveis de dificuldade foram normalizados para ajustarem-se no intervalo entre 0 e 1 nos experimentos. A Figura 4 mostra exemplos de mapas de dificuldade que correspondem a alguns quadros dos vídeos (fall e cubicle), pertencentes ao CDNet 2014. Cada pixel nesses mapas pertencem a um nível de dificuldade que varia entre 0 e 30.

(a)

(b)

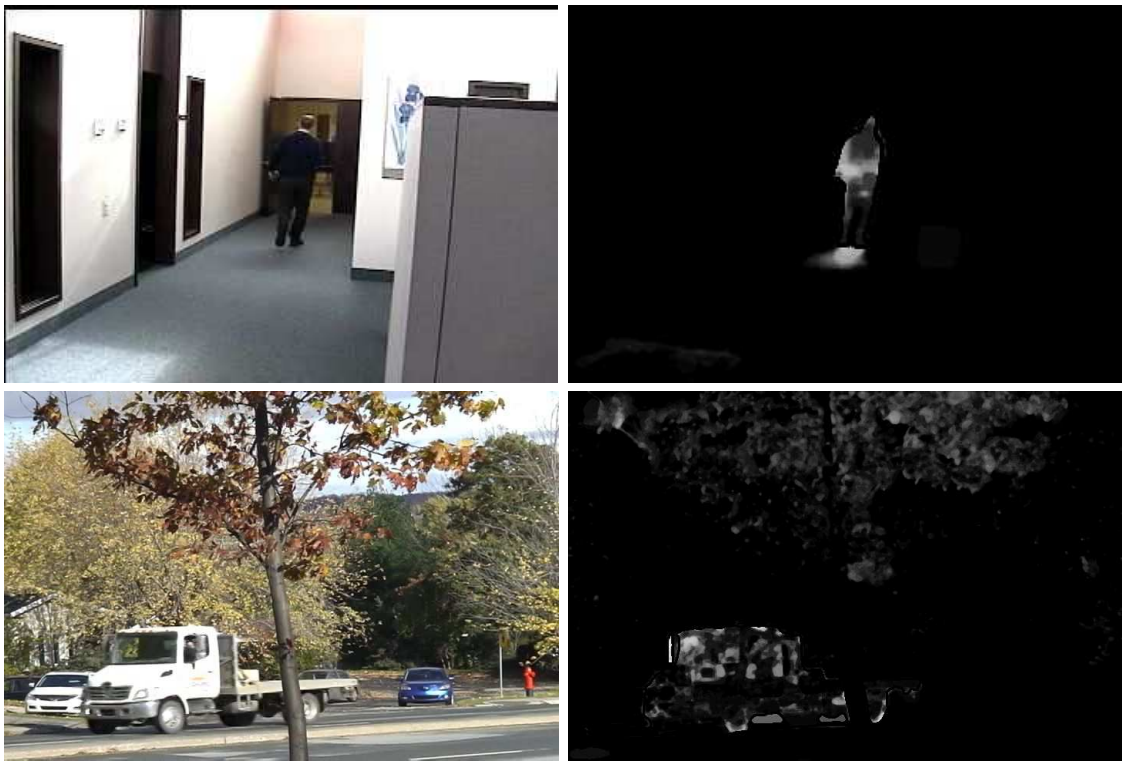

Figura 4: Exemplos de mapas de dificuldades. (a) quadro 2021 do vídeo cubicle (primeira coluna) e quadro 2021 do seu mapa de dificuldade correspondente (segunda coluna). (b) quadro 1513 do vídeo fall (primeira coluna) e quadro 1513 do mapa de dificuldade correspondente (segunda coluna)

4.2 AVALIAÇÃO DE ALGORITMOS UTILIZANDO OS MAPAS DE DIFICULDADE

Uma das formas de explorar as informações dos mapas de dificuldade é utilizá-lo como uma ferramenta que considera apenas pixels “difíceis” de serem classificados no processo de avaliação de algoritmos de detecção de mudança, conforme detalhado na seção 3.2. Essa estratégia tem como objetivo identificar os algoritmos que são capazes de classificar esses pixels, mesmo que seu desempenho geral não seja superior aos desempenhos de algoritmos do estado-da-arte. Os resultados desse tipo de avaliação permite identificar abordagens que podem ser consideradas “promissoras”, uma vez que os desafios para melhorá-las foram superados por outros algoritmos e, por esse motivo, são teoricamente menores que os desafios para melhorar algoritmos cujos erros de classificação são similares ao do estado-da-arte.

Nos experimentos realizados para avaliar a nova métrica proposta, foram

utilizados 9 algoritmos de detecção de mudança, diferentes daqueles utilizados para gerar os mapas de dificuldade que foram listados na Tabela 3. A ideia é identificar se alguns desses algoritmos são promissores Novamente, as máscaras S que contêm os resultados de segmentação dos algoritmos foram obtidos do *site* do CDNet 2014. A Tabela 4 mostra os 9 algoritmos utilizados nesta etapa.

Tabela 4: Algoritmos cujos desempenhos foram avaliados utilizando os mapas de dificuldade

Algoritmos	Referências
M4CDV2.0	Wang et al. (2018)
SWCD	Isik et al. (2018)
IUTIS-2	Bianco et al. (2017b)
AMBER	Wang e Dudek (2014)
Cascade CNN	Wang et al. (2017)
DeepBS	Babaee et al. (2018)
FgSegNet-v2	Lim e Keles (2019)
Multiscale BG Model	Lu (2014)
SemanticBGS	Braham et al. (2017)

Utilizando as Equações 9, 10, 11 e 12, os valores TP_D, TN_D, FP_D e FN_D foram obtidos para cada um dos 9 algoritmos utilizando os mapas de dificuldade D (gerados a partir dos 30 algoritmos mostrados na Tabela 3), os *ground truths* G e as matrizes R que contêm a região de interesse de cada vídeo. Em seguida, foram calculadas as métricas FPR_D, FNR_D, Pr_D, Re_D, Sp_D, PWC_D e $F1_D$ por meio das Equações 1,2, 3, 4, 6, 7 e 8 .

A Figura 5 mostra graficamente alguns exemplos da comparação entre os resultados da avaliação de desempenho utilizando apenas o *ground truth* (tradicional) e a avaliação de desempenho que utiliza os mapas de dificuldade. Cada gráfico apresenta uma combinação (categoria/métrica) em que uma métrica é calculada usando as máscaras S geradas pelos 9 algoritmos, quando executados em todos os vídeos de uma categoria.

As Figuras 5a, 5b e 5c são exemplos em que ambas as formas de avaliação não mostram diferenças significativas nos valores que representam o desempenho. Considerando a categoria Bad Weather, FNR e FNR_D são semelhantes (Figura 5a) e em relação à categoria Dynamic Background, Pr e Pr_D também são semelhantes (Figura 5b). O mesmo comportamento foi observado na Figura 5c, que considera as

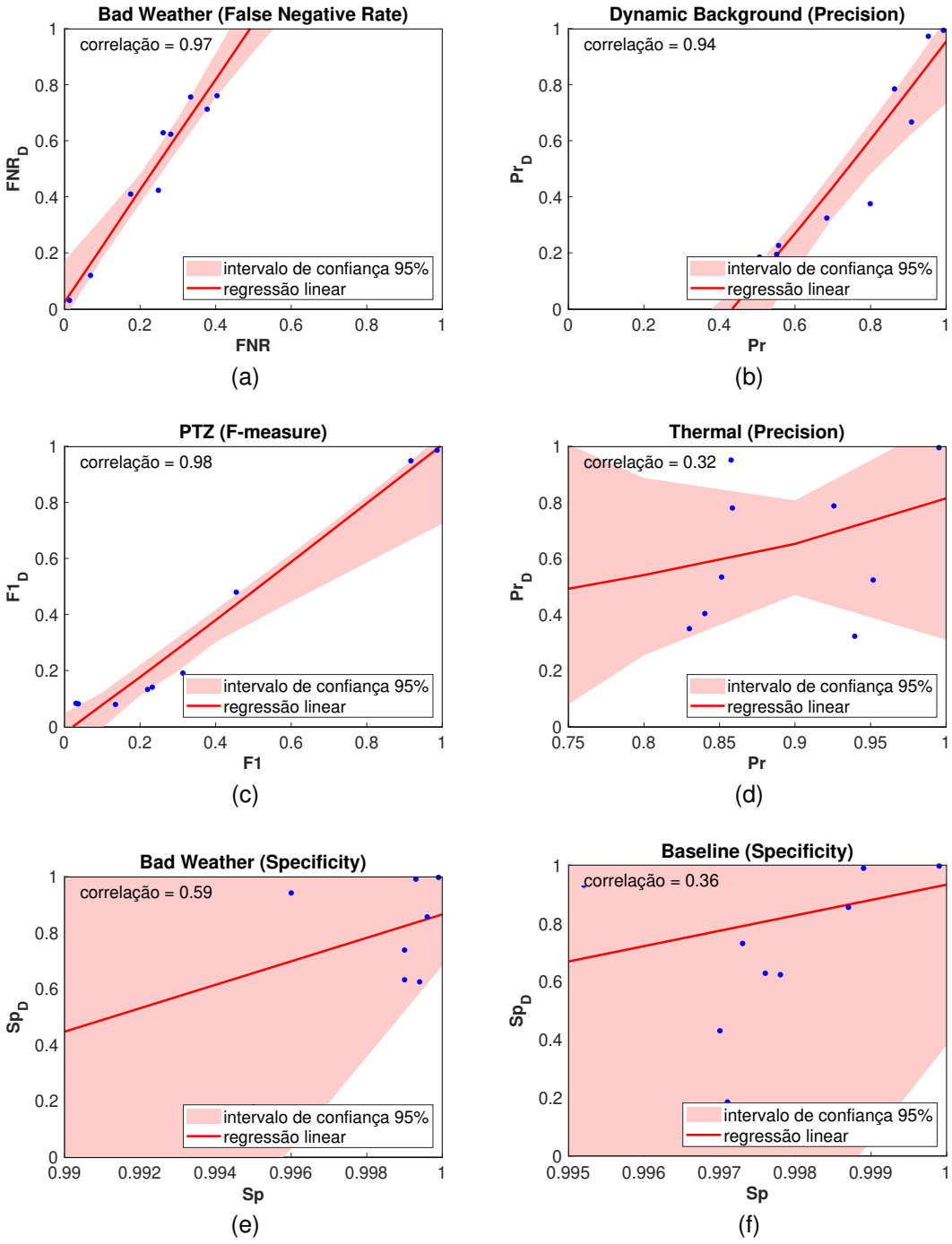

Figura 5: Exemplos de correlação entre os resultados obtidos na avaliação de desempenho comparando o uso do *ground truth* tradicional contra o uso dos mapas de dificuldade. Em (a), (b) e (c) existe uma correlação dos resultados. Em (d), (e) e (f) não há correlação dos resultados

métricas $F1$ e $F1_D$ para os vídeos da categoria PTZ.

No entanto, as métricas tradicionais, baseadas apenas no *ground truth*, e as novas métricas, baseadas nos mapas de dificuldade, não se mostraram equivalentes

em alguns casos específicos. Por exemplo, a Figura 5d mostra que as métricas Pr e Pr_D, considerando a categoria Thermal, apresentam baixa correlação (0,32). Similarmente, as métricas Sp e Sp_D apresentam baixa correlação quando considera-se a categoria Bad Weather (correlação=0,56) (Figura 5e) ou a categoria Baseline (correlação = 0,36), como pode ser observado na Figura 5f.

De acordo com o levantamento realizado em Sanches et al. (2021), a métrica F1 é utilizada para representar o desempenho na quase totalidade dos artigos científicos que apresentam novos algoritmos de detecção de mudança. Por esse motivo essa é a métrica mais importante a ser considerada na análise. Além disso, a correlação entre as métricas $F1$ e $F1_D$ considerando todos os vídeos do *dataset* pode indicar se as abordagens existentes convergem em termos de dificuldade encontradas pelos algoritmos em relação à classificação dos pixels dos quadros. A Figura 6 mostra o resultado dessa análise.

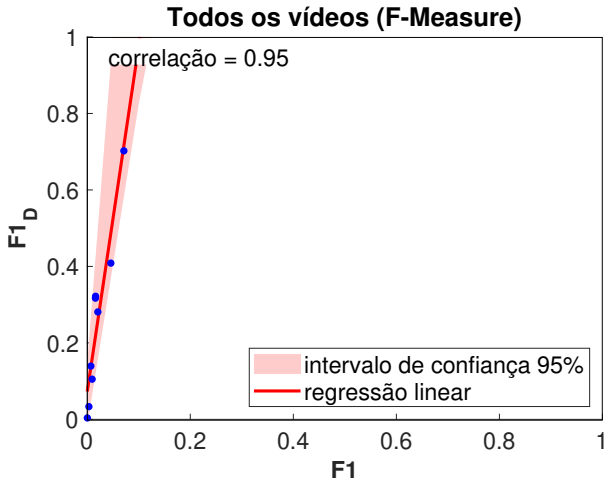

Figura 6: Correlação entre os algoritmos analisados quando avaliados pelas métricas $F1$ e $F1_D$

Como pode ser observado, existe forte correlação entre as duas métricas, quando a análise considera todos os vídeos do CDNet 2014. Isso significa que, apesar dos algoritmos se basearem nas mais diversas abordagens, não há variação nos pixels (ou grupo de pixels) que apresentam aos algoritmos maior dificuldade para classificação. Na Tabela 5 são mostradas as métricas $F1$ e $F1_D$, em ordem decrescente de valores, calculadas para os 9 algoritmos utilizados nesta etapa.

Como pode ser observado, nos algoritmos mais eficientes não há diferença

Tabela 5: Ordem dos algoritmos de acordo com as métricas $F1$ e $F1_D$

Algoritmo	F1	Algoritmo	$F1_D$
FgSegNet-v2	0,9847	FgSegNet-v2	0,9881
Cascade CNN	0,9209	Cascade CNN	0,9271
SWCD	0,7583	SWCD	0,7305
DeepBS	0,7458	DeepBS	0,6518
M4CDV2.0	0,7038	M4CDV2,0	0,5055
AMBER	0,6577	AMBER	0,4893
IUTIS-2	0,6026	SemanticBGS	0,4023
SemanticBGS	0,5813	Multiscale BG Model	0,3928
Multiscale BG Model	0,5141	IUTIS-2	0,2668

significativa nos valores de ambas as métricas e a ordem relativa do desempenho desses algoritmos é mantida. Essa ordem é perdida apenas nos algoritmos menos eficientes entre os analisados. Esses resultados indicam que os algoritmos mais recentes tendem a cometer os mesmos erros de classificação de pixels quando utilizam os vídeos do CDNet 2014 para avaliação de desempenho.

4.3 AVALIAÇÃO DE VÍDEOS UTILIZANDO OS MAPAS DE DIFICULDADE

Além da identificação de algoritmos promissores, é possível utilizar mapas de dificuldade para estimar o nível de dificuldade que um vídeo exige de um algoritmo de detecção de mudança. Um *dataset* que possui apenas vídeos em que a maioria dos quadros possui pixels fáceis de serem classificados pode não ser útil em termos práticos, pois algoritmos muito eficientes e ineficientes classificarão corretamente esses pixels. Da mesma forma, *datasets* que possuem apenas vídeos com pixels em seus quadros muito difíceis de serem classificados podem não diferenciar algoritmos ineficientes de outros que são minimamente eficientes ou possuem eficiência média.

O ideal é que um *dataset* possua vídeos que formem de uma escala de dificuldade para que cada um deles ofereça uma informação diferente sobre o desempenho de um algoritmo de detecção de mudança. É importante também que cada nível da escala contenha a mesma quantidade de vídeos. A primeira etapa para gerar um *dataset* com essas características é a identificação dos níveis de dificuldade de cada vídeo desse *dataset*.

Os mapas de dificuldade gerados utilizando a abordagem descrita na seção 3.1 mostram que os pixels marcados como nível 0 (ou seja, sem dificuldade) representam 89,48% de todos os pixels válidos dos vídeos de CDNet 2014. São considerados pixels válidos aqueles rotulados no *ground truth* como pertencentes ao elemento de interesse ou pertencentes ao fundo. Pixels rotulados como sombra, região desconhecida ou fora da região de interesse foram excluídos das análises (pixels com rótulo 0 na matriz R).

A Tabela 6 mostra os vídeos da CDNet 2014 que têm os maiores percentuais de pixels válidos rotulados no mapa com o nível de dificuldade mais baixo (nível 0). Importa ressaltar que o fato de haver muitos pixels com nível de dificuldade 0 não garante que o vídeo tenha baixo potencial de avaliação. Esse vídeo pode conter também muitos quadros formados por pixels com o nível de dificuldade mais alto ou níveis muito elevados (imediatamente inferiores ao nível mais alto de dificuldade).

Tabela 6: Vídeos com as maiores porcentagens de pixels válidos rotulados no mapa com o nível de dificuldade mais baixo (nível 0)

Video	Category	% Pixels
turbulence2	Turbulence	99,8927
pedestrians	Baseline	98,9750
fountain02	Dynamic Background	98,8563
snowFall	Bad Weather	98,6209
PETS2006	Baseline	98,3470

A complementação da análise anterior encotra-se na Figura 7, que mostra a frequência de cada nível de dificuldade (exceto nível 0) para os pixels válidos dos vídeos do *dataset* CDNet 2014. A soma dos pixels válidos com níveis de dificuldade entre 1 e 30 representam 10,52% do total de pixels válidos.

Pixels com altos níveis de dificuldade são menos frequentes nos vídeos do CDNet 2014. Uma vez que esse *dataset* se tornou a referência na área pelos pesquisadores (SANCHES et al., 2021), os desafios contidos nos seus vídeos provavelmente têm sido gradativamente superados a cada novo algoritmo apresentado.

Ainda que a análise isolada da quantidade de pixels com cada nível de dificuldade não seja conclusiva em relação ao nível de dificuldade do *dataset*, ela

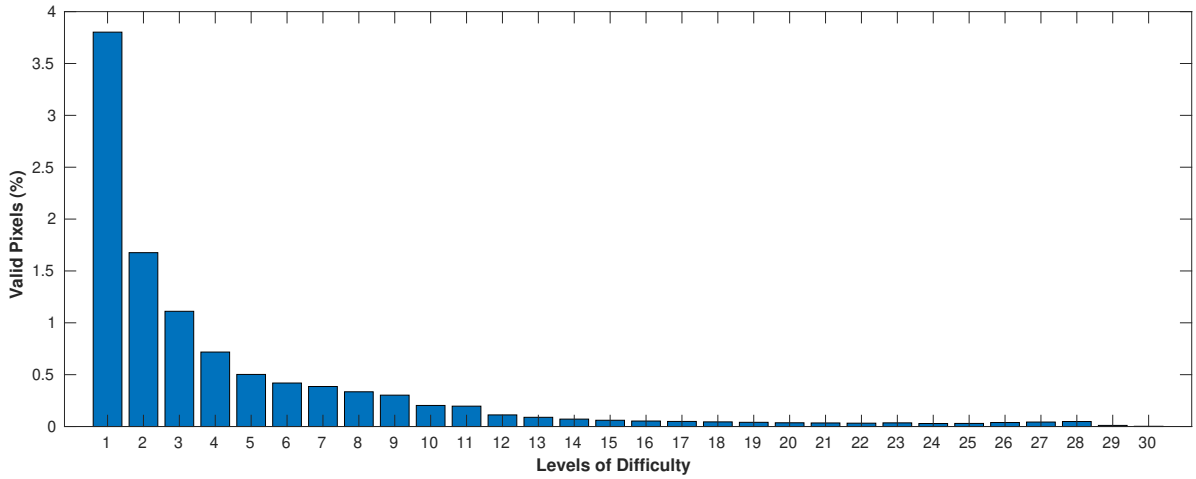

Figura 7: Frequência de ocorrência de pixels válidos para cada nível de dificuldade (exceto nível 0) obtida pelos mapas de dificuldade gerados a partir dos vídeos do CDNet 2014

pode dar indícios de que novos vídeos, com novos desafios, podem ser incluídos e, se necessário, alguns vídeos podem ser removidos do conjunto atual. As informações descritas na Tabela 7, que mostra os 5 vídeos que têm as maiores porcentuais de pixels válidos armazenados no mapa com o nível de dificuldade mais alto (nível 30), corrobora com essa constatação.

Tabela 7: Vídeos com as maiores porcentagens de pixels válidos rotulados no mapa com o nível de dificuldade mais alto (nível 30)

Vídeo	Categoria	% Pixels
busyBoulvard	Night Videos	0,0682
tramStation	Night Videos	0,0164
diningRoom	Thermal	0,0152
parking	Intermittent Object Motion	0,0108
winterStreet	Night Videos	0,0106

O fato de haver um percentual elevado de pixels com alto nível de dificuldade não necessariamente significa que grande parte do total de pixels de um quadro pode ser considerado difícil de classificar. Como as cenas exibidas nos vídeos representam cenas típicas de diversas aplicações, o número de pixels válidos pode ser reduzido no quadro. A Figura 8 mostra um exemplo de quadros onde o mapa de dificuldade contém muitos pixels com alto nível de dificuldade, embora a quantidade de pixels válidos seja pequena. Tais quadros pertencem ao vídeo parking da categoria Intermittent Object Motion.

Figura 8: Exemplos de quadros com alto nível de dificuldade: quadro original (coluna 1), *ground truth* (coluna 2) e mapa de dificuldade (coluna 3). Quadros 1306 (a), 1307 (b) e 1400 (c) do vídeo parking. O quadro 1307 contém poucos pixels válidos, como mostrado nos quadros do *ground truth*, e muitos deles são difíceis de classificar, como mostrado no mapa

Os quadros do vídeo em parking contêm poucos pixels válidos, como mostrado nos quadros do *ground truth* (column 2) das Figuras 8a-8c. No entanto, a maioria dos pixels válidos no quadro 1307 são difíceis de classificar, como mostrado nos quadros do mapa de dificuldade (coluna 3 da Figura 8b).

Essa dificuldade ocorre porque o elemento de interesse (neste caso, um veículo) começa a se mover no quadro 1307. O *ground truth* correspondente a esse quadro (coluna 2 da Figura 8b) identifica esse movimento. Pixels pertencentes a elementos de interesse em início de movimentação são difíceis de serem classificados pelos algoritmos (SANCHES et al., 2019). O mapa gerado capturou essa dificuldade (coluna 3 da Figura 8b).

A Figura 8c mostra que quando o elemento de interesse está em movimento, muitos algoritmos são capazes de classificar corretamente os pixels que pertencem a

ele. O mapa de dificuldade correspondente ao quadro 1400 (coluna 3 da Figura 8c) do vídeo parking mostra que este quadro é mais fácil de classificar quando comparado ao quadro 1307.

4.3.1 CÁLCULO DO NÍVEL DE DIFICULDADE DOS VÍDEOS DO *DATASET*

Embora os vídeos mostrados na Tabela 7 contenham pixels com níveis de dificuldade altos, esses vídeos podem não ser necessariamente os mais desafiadores para os algoritmos. Um vídeo pode não conter pixels com o nível de dificuldade máximo, no entanto, pode conter muitos pixels com outros níveis elevados (por exemplo, níveis 29, 28, 27, etc.).

A métrica proposta neste trabalho para identificar esses vídeos utiliza o valor L, definido pela Equação 15 (seção 3.3). A partir da identificação desse valor de todos os vídeos, a métrica para avaliação de vídeos seleciona um subconjunto que possui potencial de avaliação similar ao do conjunto original possui vídeos organizados na forma de uma escala de dificuldade.

A Tabela 8 mostra os vídeos do CDNet 2014 em ordem decrescente de acordo com seus níveis de dificuldade L estimados por meio da métrica proposta. Também são listados na tabela as categorias do *dataset* nas quais os vídeos pertencem. Quanto maior o valor de L, maior o nível de dificuldade do vídeo.

Nota-se que muitos vídeos do *dataset* possuem baixo nível de dificuldade, uma vez que a maioria dos algoritmos atuais são capazes de classificar corretamente os pixels dos seus quadros. Por outro lado, existem alguns vídeos em que o valor L é bastante elevado. Nesta análise não foram consideradas as categorias que o vídeo pertencem.

Alguns vídeos podem apresentar níveis de dificuldades que pouco variam entre quadros consecutivos ao passo que outros podem apresentar variação significativa no nível de dificuldade desses quadros. Para exemplificar essas situações, os valores de L foram calculados quadro a quadro nos vídeos tramstop, library, diningRoom, parking e busyBoulvard.

Os 4 primeiros são os vídeos com maior valor de L, conforme mostrado na

Tabela 8: Níveis de dificuldade L dos vídeos do *dataset* CDNet 2014

Vídeo	Categoria	L
tramstop	intermittentObjectMotion	21,6551
library	thermal	18,4753
diningRoom	thermal	6,0330
parking	intermittentObjectMotion	5,7412
intermittentPan	PTZ	5,2249
lakeSide	thermal	5,1702
copyMachine	shadow	4,9230
fall	dynamicBackground	4,7769
zoomInZoomOut	PTZ	3,9353
abandonedBox	intermittentObjectMotion	3,8619
boulevard	cameraJitter	3,8582
cubicle	shadow	3,7995
corridor	thermal	3,4242
streetLight	intermittentObjectMotion	3,2351
sofa	intermittentObjectMotion	3,2279
continuousPan	PTZ	3,0268
winterDriveway	intermittentObjectMotion	3,0037
busyBoulvard	nightVideos	2,0020
bridgeEntry	nightVideos	1,6769
boats	dynamicBackground	1,6578
traffic	cameraJitter	1,5795
office	baseline	1,4662
winterStreet	nightVideos	1,2747
turbulence0	turbulence	1,2274
tramStation	nightVideos	1,1908
bungalows	shadow	1,1717
twoPositionPTZCam	PTZ	1,1621
tunnelExit_0_35fps	lowFramerate	1,1102
skating	badWeather	1,0493
wetSnow	badWeather	0,9910
fluidHighway	nightVideos	0,8974
port_0_17fps	lowFramerate	0,8635
blizzard	badWeather	0,8509
canoe	dynamicBackground	0,8157
backdoor	shadow	0,7564
streetCornerAtNight	nightVideos	0,6812
busStation	shadow	0,5990
sidewalk	cameraJitter	0,5916
highway	baseline	0,5772
turnpike_0_5fps	lowFramerate	0,5053
overpass	dynamicBackground	0,4809
snowFall	badWeather	0,4790
turbulence1	turbulence	0,3871
peopleInShade	shadow	0,3523
badminton	cameraJitter	0,2818
park	thermal	0,2810
turbulence3	turbulence	0,2583
fountain01	dynamicBackground	0,2567
PETS2006	baseline	0,2447
tramCrossroad_1fps	lowFramerate	0,1163
fountain02	dynamicBackground	0,0538
pedestrians	baseline	0,0518
turbulence2	turbulence	0,0093

Tabela 8 e o último é o que apresenta maior quantidade de pixels com o nível mais alto de dificuldade, conforme mostrado na Tabela 7. Como cada vídeo do CDnet 2014 possui uma região de interesse temporal (alguns quadros iniciais são reservados para que o algoritmo estime um modelo do fundo), o valor L dos quadros de cada vídeo são calculados a partir do primeiro quadro dentro da região de interesse temporal.

A quantidade de quadros fora dessa região varia conforme o vídeo, por exemplo, a região de interesse temporal inicia-se no quadro 1320 do vídeo tramstop, no quadro 700 do vídeo library, no quadro 600 do vídeo diningRoom, no quadro 1100 do vídeo parking e no quadro 730 do vídeo busyBoulvard. A Figura 9 mostra os resultados dessa análise.

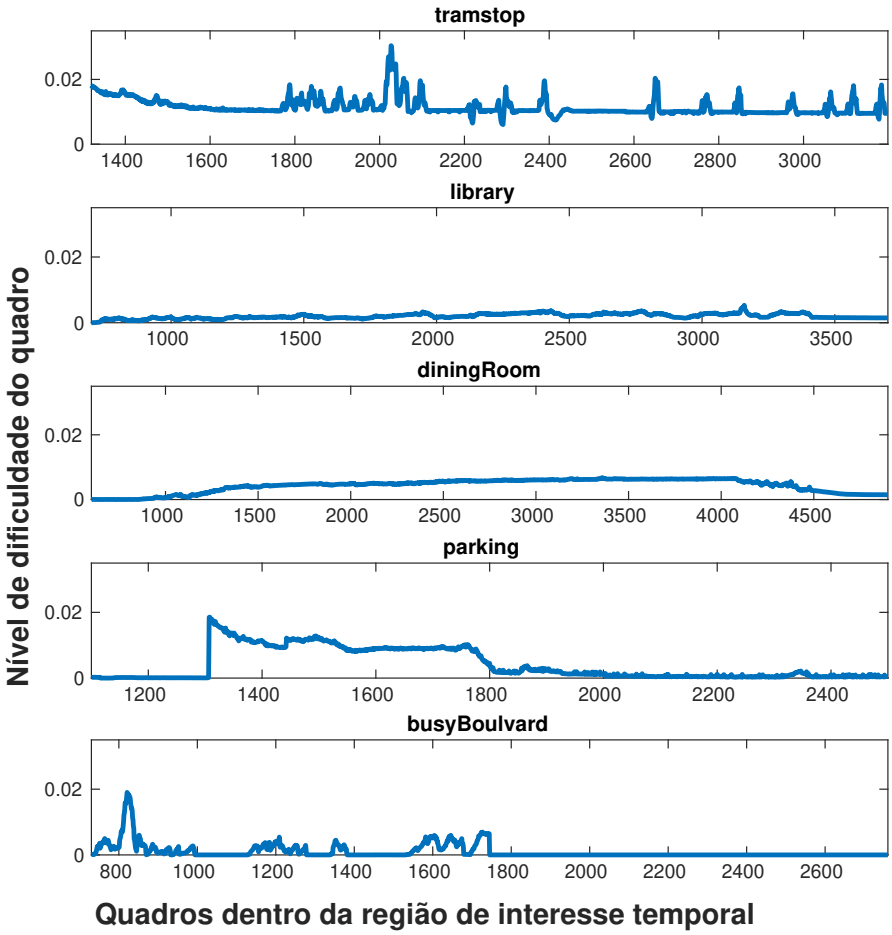

Figura 9: Níveis de dificuldade L dos quadros dos vídeos tramstop, library, diningRoom, parking e busyBoulvard

Os pixels dos vídeos busyBoulvard que possuem nível de dificuldade mais

alto aumentam os valores de L para alguns quadros desse vídeo. No entanto, em intervalos de quadros específicos, o vídeo apresenta quadros com valores muito baixos de L, o que reduz o nível de dificuldade total do vídeo. No vídeo parking, a distribuição dos níveis de dificuldade L entre os quadros é semelhante ao do vídeo busyBoulvard. No vídeo tramstop o valor L, embora sofra variações, permanece alto em todos os quadros analisados. Os vídeos diningRoom e library sofrem variações menores entre quadros consecutivos e permanecem com o nível de dificuldade sempre mais baixo em relação ao tramstop.

A Figura 10 ilustra um exemplo de variação significativa de valores L entre dois quadros consecutivos. A Figura 10a mostra os dois quadros 1744 (coluna 1) e 1745 (coluna 2) do vídeo busyBoulvard, que correspondem aos quadros 1014 e 1015 dentro da região de interesse temporal mostrada no gráfico da Figura 9. A Figura 10b mostra os mapas de dificuldade correspondentes a esses quadros (coluna 1 e coluna 2). O quadro 1745 (1015 dentro da região de interesse temporal) tem um mapa com nível de dificuldade 0 porque todos os pixels do *ground truth* estão rotulados como fora da região de interesse espacial do quadro.

(a)

(b)

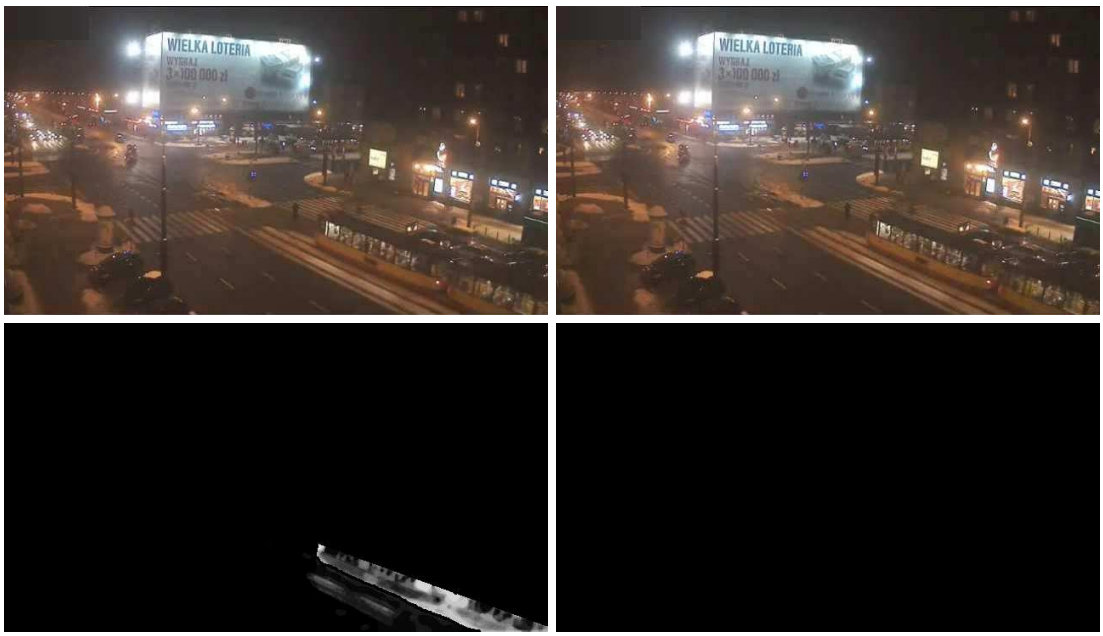

Figura 10: Exemplo de variação significativa de potencial entre dois quadros consecutivos. Em (a), Os quadros 1744 (coluna 1) e 1745 (coluna 2) e, em (b), os mapas de dificuldade desses mesmos quadros (coluna 1 e coluna 2). O mapa correspondente ao quadro 1745 contém o nível de dificuldade 0 (0%) porque todos os pixels do *ground truth* são rotulados como "fora da região de interesse temporal"

4.3.2 SELEÇÃO DE UM SUBCONJUNTO DE VÍDEOS REPRESENTATIVO DO *DATASET*

Uma vez estimados os níveis de dificuldade L dos vídeos, a etapa final consiste na seleção de um subconjunto representativo, que possua o mesmo potencial de avaliação do conjunto de vídeos original e que contenha vídeos organizados na forma de uma escala de dificuldade. Considerando os valores obtidos, a seleção dos vídeos que compõem o subconjunto pode ser realizada por meio do Algoritmo 2. Os resultados mostraram que a abordagem proposta selecionou 36 dos 53 vídeos do CDNet 2014. Esses vídeos são listados na Tabela 9, acompanhados dos seus respectivos níveis de dificuldade L.

Tabela 9: Vídeos representativos selecionados pela abordagem proposta

Vídeos	L	Vídeos	L
tramstop	21,6551	winterStreet	1,2747
diningRoom	6,0330	turbulence0	1,2274
library	18,4753	twoPositionPTZCam	1,1621
parking	5,7412	tunnelExit_0_35fps	1,1102
intermittentPan	5,2249	skating	1,0493
lakeSide	5,1702	wetSnow	0,9910
copyMachine	4,9230	fluidHighway	0,8974
fall	4,7769	blizzard	0,8509
zoomInZoomOut	3,9353	canoe	0,8157
boulevard	3,8582	backdoor	0,7564
cubicle	3,7995	streetCornerAtNight	0,6812
corridor	3,4242	highway	0,5772
sofa	3,2279	snowFall	0,4790
winterDriveway	3,0037	turbulence1	0,3871
busyBoulvard	2,0020	peopleInShade	0,3523
boats	1,6578	PETS2006	0,2447
traffic	1,5795	tramCrossroad_1fps	0,1163
office	1,4662	pedestrians	0,0518

Para comparar o potencial dos dois conjuntos, foram escolhidos 12 algoritmos cujos desempenhos foram avaliados em ambos (*dataset* original e subconjunto representativo). Além dos utilizados para validar a métrica que avalia algoritmos, outros 3 algoritmos foram incluídos, são eles: BSGAN (ZHENG et al., 2020), SOBS_CF (MADDALENA; PETROSINO, 2010) e DCB (KRUNGKAEW; KUSAKUNNIRAN, 2016). A Tabela 10 mostra os algoritmos utilizados nesta etapa.

Tabela 10: Algoritmos cujos desempenhos foram avaliados utilizando os vídeos representativos

Algoritmo	Referência
M4CDV2.0	Wang et al. (2018)
SWCD	Isik et al. (2018)
IUTIS-2	Bianco et al. (2017b)
AMBER	Wang e Dudek (2014)
Cascade CNN	Wang et al. (2017)
DeepBS	Babaee et al. (2018)
FgSegNet-v2	Lim e Keles (2019)
Multiscale BG Model	Lu (2014)
SemanticBGS	Braham et al. (2017)
BSGAN	Zheng et al. (2020)
SOBS_CF	Maddalena e Petrosino (2010)
DCB	Krungkaew e Kusakunniran (2016)

Ainda que as máscaras S desses algoritmos não estejam disponíveis no *site* do CDNet 2014 (*links* quebrados), seus valores de desempenho FP, FN, TP e TN podem ser obtidos para cada vídeo no *site* do CDNet 2014, o que permite calcular a métrica $F1$. Esses valores são suficientes para as análises realizadas nesta etapa. A métrica $F1$ foi considerada como valor de desempenho dos algoritmos nesta análise uma vez que é a mais utilizada pelos pesquisadores da área (SANCHES et al., 2021).

Para identificar se o potencial do conjunto representativo é similar ao potencial do conjunto original, os desempenhos dos 12 algoritmos foram avaliados utilizando os dois conjuntos de vídeos. A Tabela 11 mostra o resultado dessa análise. Como pode ser observado, a ordem relativa dos algoritmos é mantida nas duas avaliações e, além disso, os valores de desempenho (métrica $F1$) de um mesmo algoritmo não apresentam variações significativas nas duas avaliações.

Os resultados da análise indicam que o CDNet 2014 possui muitos vídeos com níveis de dificuldade similares. Muitos desses vídeos não contribuem para que o conjunto avalie o desempenho de um algoritmo de forma precisa. Por meio da abordagem proposta, foi possível reduzir o número total de vídeos (de 53 para 36), mantendo-se o mesmo potencial de avaliação do *dataset*.

Tabela 11: Desempenhos dos 12 algoritmos avaliados utilizando o conjunto de vídeos original e conjunto representativo

Todos os vídeos		**Subconjunto representativo**	
Algoritmo	$F1$	**Algoritmo**	$F1$
FgSegNet-v2	0,9857	FgSegNet-v2	0,9906
BSGAN	0,9350	BSGAN	0,9460
Cascade CNN	0,9200	Cascade CNN	0,9252
SemanticBGS	0,7909	SemanticBGS	0,7862
SWCD	0,7602	SWCD	0,7687
DeepBS	0,7467	DeepBS	0,7579
M4CDV2.0	0,7029	M4CDV2.0	0,7050
AMBER	0,6644	AMBER	0,6685
IUTIS-2	0,5975	IUTIS-2	0,5944
SOBS_CF	0,5944	SOBS_CF	0,5874
Multiscale BG Model	0,5214	Multiscale BG Model	0,5229
DCB	0,4115	DCB	0,4469

5 CONCLUSÕES

A avaliação do desempenho de um algoritmo de detecção de mudança deve mostrar a superioridade desse algoritmo em relação ao estado da arte. Avaliar um algoritmo consiste na sua execução para segmentar os vídeos de um *dataset* e na a comparação dos resultados com um *ground truth*. Para que o desempenho dos algoritmos sejam obtidos de forma precisa, o *dataset* utilizado deve conter vídeos com diferentes níveis de dificuldade, para que cada vídeo produza uma informação diferente sobre o desempenho do algoritmo avaliado.

Neste trabalho foram apresentadas duas métricas, uma para obter o desempenho de algoritmos de detecção de mudança em relação a um mapa de dificuldade e outra para estimar o nível de dificuldade de cada vídeo de um *dataset*. Um vez obtidos esses níveis de dificuldade, foi desenvolvido um método para selecionar um subconjunto de vídeos representativo de um *dataset*. Esse conjunto possui potencial de avaliação similar ao do *dataset* completo, ainda que contenha um número menor de vídeos.

Os resultados obtidos da aplicação da métrica para avaliação de algoritmos mostraram que as soluções utilizadas atualmente, apesar da diversidade de abordagens, cometem erros semelhantes quando segmentam os mesmos vídeos. Em outras palavras, os erros de classificação cometidos pelos algoritmos normalmente ocorrem nos mesmos pixels.

Em relação à métrica que estima o nível de dificuldade de um vídeo, os resultados da sua aplicação nos vídeos do *dataset* CDNet 2014 permitiram que um subconjunto de vídeos representativo fosse identificado a partir dessa informação. O método para selecionar esse subconjunto, que foi desenvolvido neste trabalho, mostrou-se eficiente uma vez que o subconjunto apresentou o mesmo potencial de

avaliação do conjunto original.

Como perspectivas de trabalhos futuros, pretende-se desenvolver métricas para estimar níveis de dificuldade de vídeos de *datasets* utilizados para avaliar algoritmos que resolvem outros problemas na área de visão computacional, como o rastreamento de pessoas, a detecção de acidentes em rodovias ou a detecção de objetos abandonados locais públicos.

REFERÊNCIAS

ALLEBOSCH, G.; DEBOEVERIE, F.; VEELAERT, P.; PHILIPS, W. Efic: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints. In: BATTIATO, S.; BLANC-TALON, J.; GALLO, G.; PHILIPS, W.; POPESCU, D.; SCHEUNDERS, P. (Ed.). **Advanced Concepts for Intelligent Vision Systems**. Cham: Springer International Publishing, 2015. p. 130–141. ISBN 978-3-319-25903-1.

ALLEBOSCH, G.; HAMME, D. V.; DEBOEVERIE, F.; VEELAERT, P.; PHILIPS, W. C-efic: Color and edge based foreground background segmentation with interior classification. In: BRAZ, J.; PETTRÉ, J.; RICHARD, P.; KERREN, A.; LINSEN, L.; BATTIATO, S.; IMAI, F. (Ed.). **Computer Vision, Imaging and Computer Graphics Theory and Applications**. Cham: Springer International Publishing, 2016. p. 433–454. ISBN 978-3-319-29971-6.

BABAEE, M.; DINH, D. T.; RIGOLL, G. A deep convolutional neural network for video sequence background subtraction. **Pattern Recognition**, v. 76, p. 635 – 649, 2018. ISSN 0031-3203.

BENEZETH, Y.; JODOIN, P.-M.; EMILE, B.; LAURENT, H.; ROSENBERGER, C. Comparative study of background subtraction algorithms. **Journal of Electronic Imaging**, SPIE, v. 19, n. 3, p. 1 – 12, 2010.

BIANCO, S.; CIOCCA, G.; SCHETTINI, R. Combination of video change detection algorithms by genetic programming. **IEEE Transactions on Evolutionary Computation**, v. 21, n. 6, p. 914–928, Dec 2017. ISSN 1941-0026.

BIANCO, S.; CIOCCA, G.; SCHETTINI, R. How far can you get by combining change detection algorithms? In: BATTIATO, S.; GALLO, G.; SCHETTINI, R.; STANCO, F. (Ed.). **Image Analysis and Processing - ICIAP 2017**. Cham: Springer International Publishing, 2017. p. 96–107. ISBN 978-3-319-68560-1.

BRAHAM, M.; PIéRARD, S.; DROOGENBROECK, M. V. Semantic background subtraction. In: **2017 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2017. p. 4552–4556.

CHEN, Y.; WANG, J.; LU, H. Learning sharable models for robust background subtraction. In: **2015 IEEE International Conference on Multimedia and Expo (ICME)**. [S.l.: s.n.], 2015. p. 1–6. ISSN 1945-788X.

ELGAMMAL, A.; HARWOOD, D.; DAVIS, L. Non-parametric model for background subtraction. In: VERNON, D. (Ed.). **Computer Vision — ECCV 2000**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 751–767. ISBN 978-3-540-45053-5.

FISHER, R. **CAVIAR Test Case Scenarios**. 2019. http://groups.inf.ed.ac.uk/vision/CAVIAR Accessed 24 Sep 2019.

GOYETTE, N.; JODOIN, P. M.; PORIKLI, F.; KONRAD, J.; ISHWAR, P. Changedetection.net: A new change detection benchmark dataset. In: **2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2012. p. 1–8. ISSN 2160-7508.

GREGORIO, M. D.; GIORDANO, M. Change detection with weightless neural networks. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 409–413. ISSN 2160-7508.

GREGORIO, M. D.; GIORDANO, M. Wisardrp for change detection in video sequences. In: **25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2017)**. [S.l.: s.n.], 2017. p. 453–458. ISSN 978-287587039-1.

ISIK, S.; ÖZKAN, K.; GÜNAL, S.; GEREK, O. N. Swcd: a sliding window and self-regulated learning-based background updating method for change detection in videos. **Journal of Electronic Imaging**, v. 27, n. 2, p. 1–11, 2018.

JIANG, S.; LU, X. Wesambe: A weight-sample-based method for background subtraction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 28, n. 9, p. 2105–2115, Sep. 2018.

Kalsotra, R.; Arora, S. A comprehensive survey of video datasets for background subtraction. **IEEE Access**, v. 7, p. 59143–59171, 2019. ISSN 2169-3536.

KRUNGKAEW, R.; KUSAKUNNIRAN, W. Foreground segmentation in a video by using a novel dynamic codebook. In: **2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)**. [S.l.: s.n.], 2016. p. 1–6.

LEE, S.-h.; LEE, G.-c.; YOO, J.; KWON, S. Wisenetmd: Motion detection using dynamic background region analysis. **Symmetry**, v. 11, n. 5, p. 1–15, 2019. ISSN 2073-8994.

LI, L.; HUANG, W.; GU, I. Y.-H.; TIAN, Q. Statistical modeling of complex backgrounds for foreground object detection. **Trans. Img. Proc.**, IEEE Press, Piscataway, NJ, USA, v. 13, n. 11, p. 1459–1472, nov. 2004. ISSN 1057-7149.

LIANG, D.; KANEKO, S.; HASHIMOTO, M.; IWATA, K.; ZHAO, X. Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes. **Pattern Recognition**, v. 48, n. 4, p. 1374 – 1390, 2015. ISSN 0031-3203.

LIM, L. A.; KELES, H. Y. Foreground segmentation using convolutional neural networks for multiscale feature encoding. **Pattern Recognition Letters**, v. 112, p. 256 – 262, 2018. ISSN 0167-8655.

LIM, L. A.; KELES, H. Y. Learning multi-scale features for foreground segmentation. **Pattern Analysis and Applications**, Aug 2019. ISSN 1433-755X.

LU, X. A multiscale spatio-temporal background model for motion detection. In: **2014 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2014. p. 3268–3271. ISSN 2381-8549.

MADDALENA, L.; PETROSINO, A. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. **Neural Computing and Applications**, v. 19, n. 2, p. 179–186, Mar 2010. ISSN 1433-3058.

MADDALENA, L.; PETROSINO, A. The sobs algorithm: What are the limits? In: **2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2012. p. 21–26. ISSN 2160-7516.

MARTINS, I.; CARVALHO, P.; CORTE-REAL, L.; ALBA-CASTRO, J. L. Bmog: Boosted gaussian mixture model with controlled complexity. In: ALEXANDRE, L. A.; SÁNCHEZ, J. S.; RODRIGUES, J. M. F. (Ed.). **Pattern Recognition and Image Analysis**. Cham: Springer International Publishing, 2017. p. 50–57. ISBN 978-3-319-58838-4.

MICROSOFT CORPORATION. **Test Images for Wallflower Paper**. 2019. https://www.microsoft.com/en-us/download/details.aspx?id=54651. Accessed 9 Aug 2019.

MIRON, A.; BADII, A. Change detection based on graph cuts. In: **2015 International Conference on Systems, Signals and Image Processing (IWSSIP)**. [S.l.: s.n.], 2015. p. 273–276. ISSN 2157-8702.

OPENCV TEAM. **OpenCV**. 2019. https://opencv.org/. Accessed 24 Sep 2019.

RAMíREZ-ALONSO, G.; CHACON-MURGUIA, M. I. Auto-adaptive parallel som architecture with a modular analysis for dynamic object segmentation in videos. **Neurocomputing**, v. 175, p. 990 – 1000, 2016. ISSN 0925-2312.

RUSSEL, J.; COHN, R. **Interquartile Range**. [S.l.]: Tbilisi State University, 2013. ISBN 9785510797251.

SAJID, H.; CHEUNG, S. S. Universal multimode background subtraction. **IEEE Transactions on Image Processing**, v. 26, n. 7, p. 3249–3260, July 2017. ISSN 1941-0042.

SANCHES, S. R. R.; OLIVEIRA, C.; SEMENTILLE, A. C.; FREIRE, V. Challenging situations for background subtraction algorithms. **Applied Intelligence**, v. 49, n. 5, p. 1771–1784, May 2019. ISSN 1573-7497.

SANCHES, S. R. R.; SEMENTILLE, A. C.; AGUILAR, I. A.; FREIRE, V. Recommendations for evaluating the performance of background subtraction algorithms for surveillance systems. **Multimedia Tools and Applications**, Springer, v. 80, n. 3, p. 4421–4454, 2021.

SEDKY, M.; MONIRI, M.; CHIBELUSHI, C. C. Spectral-360: A physics-based technique for change detection. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 405–408. ISSN 2160-7508.

SOBRAL, A.; VACAVANT, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. **Computer Vision and Image Understanding**, Elsevier Inc., v. 122, p. 4–21, 2014. ISSN 10773142.

ST-CHARLES, P.; BILODEAU, G.; BERGEVIN, R. A self-adjusting approach to change detection based on background word consensus. In: **2015 IEEE Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2015. p. 990–997. ISSN 1550-5790.

ST-CHARLES, P.; BILODEAU, G.; BERGEVIN, R. Subsense: A universal change detection method with local adaptive sensitivity. **IEEE Transactions on Image Processing**, v. 24, n. 1, p. 359–373, Jan 2015. ISSN 1941-0042.

STAUFFER, C.; GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. In: **Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)**. [S.l.: s.n.], 1999. v. 2, p. 246–252 Vol. 2. ISSN 1063-6919.

TOYAMA, K.; KRUMM, J.; BRUMITT, B.; MEYERS, B. Wallflower: principles and practice of background maintenance. In: **Proceedings of the Seventh IEEE International Conference on Computer Vision**. [S.l.: s.n.], 1999. v. 1, p. 255–261 vol.1.

UNIVERSITÉ DE SHERBROOKE. **ChangeDetection.NET – A video database for testing change detection algorithms**. 2019. http://www.changedetection.net. Accessed 22 July 2018.

UNIVERSITY OF NAPLES PARTHENOPE. **SceneBackgroundModeling.net.NET – A video database for testing background estimation algorithms**. 2019. http://scenebackgroundmodeling.net. Accessed 24 July 2019.

VACAVANT, A.; CHATEAU, T.; WILHELM, A.; LEQUIÈVRE, L. A benchmark dataset for outdoor foreground/background extraction. In: ____. **Computer Vision - ACCV 2012 Workshops: ACCV 2012 International Workshops, Daejeon, Korea, November 5-6, 2012, Revised Selected Papers, Part I**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 291–300. ISBN 978-3-642-37410-4.

VARADARAJAN, S.; MILLER, P.; ZHOU, H. Spatial mixture of gaussians for dynamic background modelling. In: **2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance**. [S.l.: s.n.], 2013. p. 63–68. ISSN null.

VARGHESE, A.; G, S. Sample-based integrated background subtraction and shadow detection. **IPSJ Transactions on Computer Vision and Applications**, v. 9, n. 1, p. 25, Dec 2017. ISSN 1882-6695.

WANG, B.; DUDEK, P. A fast self-tuning background subtraction algorithm. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 401–404. ISSN 2160-7508.

WANG, K.; GOU, C.; WANG, F.-Y. **M4CD: A Robust Change Detection Method for Intelligent Visual Surveillance**. 2018. https://arxiv.org/abs/1802.04979. Cornell University. Accessed 12 Nov 2019.

WANG, R.; BUNYAK, F.; SEETHARAMAN, G.; PALANIAPPAN, K. Static and moving object detection using flux tensor with split gaussian models. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 420–424. ISSN 2160-7508.

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PPGI
programa de pós-graduação
em informática

@ppgicp

www ppgi-cp@utfpr.edu.br
portal.utfpr.edu.br/cp/ppgi